PRE-APPEAL BRIEF REQUEST FOR REVIEW

Docket Number (Optional)
CE11193JI210

I hereby certify that this correspondence is being electronically transmitted on the date listed below [(37 CFR 1.8(a)].	Application Number	Filed
on: September 13, 2007	10/643,327	August 19, 2003
Signature /Larry G. Brown/	First Named Inventor: Jean Khawand	
Larry G. Brown (Typed or printed name)	Art Unit: 2182	Examiner: Sorrell, Eron J

Applicant requests review of the final rejection in the above-identified application. No amendments are being filed with this request.

This request is being filed with a notice of appeal.

The review is requested for the reason(s) stated on the attached sheets(s).
Note: No more than five (5) pages may be provided.

I am the

☐ applicant inventor.

☐ assignee of record of the entire interest.
See 37 CFR 3.71. Statement under 37 CFR 3.73(b) is enclosed. (Form PTO/SB/96)

☒ attorney or agent of record.
Registraton number 45,834

☐ attorney or agent acting under 37 CFR 1.34.
Registration number if acting under 37 CFR 1.34: _____

/Larry G. Brown/
Signature

Larry G, Brown
Typed or printed name

(954) 723-6449
Telephone number

September 13, 2007
Date

NOTE: Signatures of all the inventors or assignees or record of the entire interest or their representative(s) are required. Submit multiple forms if more than one signature is required, se below*

☒ *Total of 1 forms are submitted.

(SB/33 (07-05)

UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICANT(S):	Jean Khawand	GROUP ART UNIT:	2182
APPLN. NO.:	10/643,327	EXAMINER:	Sorrell, Eron J
FILED:	August 19, 2003	Confirmation No.	4001
TITLE:	METHOD AND APPARATUS FOR PROVIDING INTERPROCESSOR COMMUNICATIONS USING SHARED MEMORY		

CERTIFICATE UNDER 37 CFR 1.8(a)	
I hereby certify that this correspondence is being electronically transmitted on the date listed below:	
Date:	September 13, 2007
Signature	/Larry G. Brown/
Typed or printed name:	Larry G. Brown

PRE-APPEAL BRIEF REQUEST FOR REVIEW

Mail Stop: **AF**
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

Applicants request review of the Non-Final Office Action mailed April 13, 2007 relating to the above-identified application in furtherance of the Notice of Appeal filed on September 13, 2007. Although this pre-appeal brief stems from a non-final office action, the claims at issue have been rejected at least twice.

Claims 1-19 remain pending in the application. The most recent copy of the claims can be found in Applicants' Amendment of February 13, 2007. In the Office Action, claims 1-3, 7-11, 15, 18 and 19 were rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent No. 6,912,716 to Johanson, et al. (Johanson) in view of U.S. Patent No. 6,131,113 to Ellsworth, et al. (Ellsworth). Claims 4-6, 12-14, 16 and 17

were rejected under 35 U.S.C. 103(a) as being unpatentable over Johanson in view of Ellsworth and further in view of U.S. Patent No. 6,823,511 to McKenney, et al. (McKenney).

Independent Claims 1, 11 and 15

Independent claim 1 recites the limitation that the shared memory includes the transmit memories of both the first and second processors and that the first processor is a master processor that manages the shared memory. It is inherent, then, that the first processor manages the transmit memories of the first and second processors, as one of skill in the art would appreciate. Independent claims 11 and 15 contain similar limitations.

Applicants submit that Johanson (or any other prior art reference) does not disclose such a feature. Johanson calls for a shared memory that is separated into three main portions: (1) a first processor to second processor fixed portion; (2) a second processor to first processor fixed portion; and (3) an unallocated dynamic portion (see col. 4, lines 8-11). The first processor has write access in an allocation starting at the bottom of the shared memory – or portion (1) above- and filling upwards, while the second processor has write access in an allocation starting at the top of the shared memory – portion (2) above – and filling downward (see col. 5, lines 13-18). Depending on the size of the message, either processor can dynamically allocate messages in the unallocated dynamic portion, with the first processor allocating from lower to higher addresses and the second processor in the opposite direction (see col. 4, lines 28-37). Both processors have read access to all portions of the shared memory (see col. 5, lines 18-21). Johanson goes on to describe one possible implementation in which the first processor has access to the dynamic unallocated portion, while the only way the

second processor would be allowed to write to this unallocated portion is by requesting write access from the first processor (see col. 6, lines 44-50).

In view of this description, it is clear that neither the first processor nor the second processor is a master processor that manages the shared memory, i.e., a processor that completely manages the shared memory that includes the transmit memories of the first and second processors. In other words, while both processors may have read access to all the shared memory, neither has write access to both fixed portions of the shared memory. Even when one processor is designated as the controller of the unallocated portion, that processor still does not have write access (i.e., allocation) to the fixed portion assigned to the other processor. This scheme results in a forced static division of shared memory between the two processors, a disadvantage that the presently claimed subject matter seeks to avoid (see page 1 of the present application).

Each of independent claims 1, 11 and 15 also recites the limitation of the first processor sending a message buffer pointer to the second processor that directs the second processor to the message buffer. Applicants submit that Ellsworth does not describe such a feature.

As explained in column 7, lines 23-49 of Ellsworth, the first processor places in a first data element of the resource queue a pointer to an available section of the shared resource. The first processor will then increment a tail pointer for the resource queue. The first processor will continue to place the pointers to the available section in the resource queue and to increment the tail pointer until it determines – by detecting that the tail pointer is pointing to the queue end - that the resource queue is full.

In addition and as described in column 7, line 50 to column 8, line 31, the second processor, will fetch a pointer to the next available portion of the shared resource. The second processor will then appropriately adjust the head pointer and transmit a "resource consumed" event message to the mailbox message subsystem associated with the first processor. The resource consumed message notifies the first processor that a portion of the shared resource was allocated and that an additional available shared resource section can be placed in the resource queue. As described in column 8, line 51 to column 9, line 3, in response to the resource consumed message, the first processor stores a pointer to an available section of the of the shared resource in the resource queue location pointed to by the tail pointer. The first processor will also set the tail pointer in the appropriate position.

The claimed limitation that the first processor sends a message buffer pointer to the second processor that directs the second processor to the message buffer is important because it enables the first processor to actively manage the shared memory, while at the same time it eliminates the requirement that the second processor actively manage the shared memory. Specifically, the first processor can receive a request from the second processor and can reserve or allocate a known portion of the shared memory for the second processor, the location of which – in accordance with the claim language - can be forwarded to the second processor. In direct contrast, the second processor in Ellsworth does not receive such a message from the first processor, and as a result, the second processor must actively search for available portions of the shared resource. That is, the first processor in Ellsworth merely provides pointers to the resource queue, not to the second processor.

Moreover, because the first processor in Ellsworth does not send message buffer pointers to the second processor, the first processor is unaware of the actual memory needs of the second processor. In other words, in Ellsworth, the first processor blindly makes portions of the shared resource available to the second processor. This method of providing portions of a shared resource is inefficient management of the shared resource. Even worse, this inefficiency is compounded as the number of processors in the system to which the first processor must provide portions of the shared resource increases. Such is not the case with the present invention in view of the first processor knowing the memory needs of any number of processors and the first processor's ability to signal these multiple processors with address pointers.

Conclusion

In view of the above, Applicants contend that the claims are patentable over the cited prior art references. Reconsideration and withdrawal of the rejection of the claims is respectfully requested. Passing of this case is now believed to be in order, and a Notice of Allowance is earnestly solicited.

The Commissioner is hereby authorized to charge any necessary fee, or credit any overpayment, to Motorola, Inc. Deposit Account No. 50-2117.

Respectfully submitted,

Date: September 13, 2007

By: /Larry G. Brown/
Larry G. Brown
Attorney of Record
Reg. No.: 45,834

SEND CORRESPONDENCE TO:
Motorola, Inc.
Law Department – MD 1610
8000 W. Sunrise Blvd.
Plantation, FL 33322

Customer Number: 24273

Tel: (954) 723-4295 direct line
Tel: (954) 723-6449 main line

Fax No.: (954) 723-3871